Comment couper la poire en deux ?

Épreuve pratique d'algorithmique et de programmation
Concours commun des Écoles normales supérieures

Durée de l'épreuve : 3 heures 30 minutes

Juin/Juillet 2021

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0
à l'emplacement prévu sur votre fiche réponse

Important.

Il vous a été donné un numéro u_0 qui servira d'entrée à vos programmes. Les réponses attendues sont généralement courtes et doivent être données sur la fiche réponse fournie à la fin du sujet. À la fin du sujet, vous trouverez en fait deux fiches réponses. La première est un exemple des réponses attendues pour un $\widetilde{u_0}$ particulier (précisé sur cette même fiche et que nous notons avec un tilde pour éviter toute confusion !). Cette fiche est destinée à vous aider à vérifier le résultat de vos programmes en les testant avec $\widetilde{u_0}$ au lieu de u_0. Vous indiquerez vos réponses (correspondant à votre u_0) sur la seconde et vous la remettrez à l'examinateur à la fin de l'épreuve.

En ce qui concerne la partie orale de l'examen, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures !
Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, par exemple : $O(n^2)$, $O(n \log n)$,...
Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de ***tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie en annexe***.
Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

L'objectif général de ce sujet est, pour un graphe donné, de le partitionner en deux parties de poids équilibrés tout en modérant le nombre d'arêtes à cheval entre les deux parties.

1 Préliminaires

On rappelle que pour deux entiers naturels a et b, $a \bmod b$ désigne le reste de la division entière de a par b.
Dans tout le sujet les listes sont indexées à partir de 0.

1.1 Suite de nombres pseudo-aléatoires

On fixe $M = 2^{31} - 1 = 2\,147\,483\,647$.
On définit la suite $\overline{u}(n)$ par récurrence :

$$\overline{u}(0) = u_0, \quad \forall n \in \mathbb{N}, \overline{u}(n+1) = (16\,807 \times \overline{u}(n) + 17) \bmod M$$

u_0 vous ayant été donné, et doit être reporté sur votre fiche réponse.
On définit alors la suite

$$u(n) = \overline{u}(n \bmod 999\,983)$$

Question 1 *Calculer* $u(n) \bmod 997$ *pour*
a) $n = 16$, ***b)*** $n = 1024$, ***c)*** $n = 1\,000\,000$.

1.2 Somme de contrôle de listes de nombres

On définit l'opérateur de somme de contrôle :

$$S(x, y) = (2 \times x + y) \bmod 997$$

On définit la *somme de contrôle* d'une liste de nombres par récurrence :

$$C([]) = 0, \quad \forall n \in \mathbb{N}\ C([x_1, x_2, ..., x_{n+1}]) = S(C([x_1, x_2, ..., x_n]), x_{n+1})$$

Question 2 *Calculer*
a) $S(u(16), u(17))$, ***b)*** $C([u(100), u(101), ..., u(1\,000)])$,
c) $C([u(999\,900), u(999\,901), ..., u(1\,000\,000)])$.

Question à développer pendant l'oral 1 *Expliquez votre implémentation et sa complexité.*

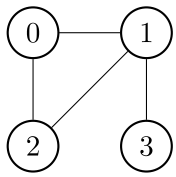
1.3 Génération de graphe aléatoire

On souhaite ici générer des graphes pseudo-aléatoirement. Tous les graphes manipulés seront simples, i.e. sans arête d'un sommet à lui-même (boucle) ni arêtes multiples pour une même paire de sommets.
Pour un n et un a donné, on note G_n^a le graphe non orienté ayant n sommets $\{s_0, s_1, ..., s_{n-1}\}$ et dont l'ensemble des arêtes est

$$\begin{aligned}\{\ & (s_{(u(a+2i)) \bmod n}, s_{(u(a+2i+1)) \bmod n}) \mid \\ & i \in [\![0, n \cdot \lfloor\sqrt{n}\rfloor - 1]\!] \\ & \text{et } (u(a+2i)) \bmod n \neq (u(a+2i+1)) \bmod n\ \}\end{aligned}$$

Il n'y a donc pas de boucle, et les doublons étant ignorés, il n'y a pas d'arêtes multiples non plus.

Par exemple, pour $\widetilde{u_0}$, $\widetilde{G_4^{16}}$ a donc pour ensemble d'arêtes $\{(0,2),(1,0),(1,3),(2,1)\}$, car la boucle $(0,0)$ est exclue, et l'arête $(0,1)$ est confondue avec l'arête $(1,0)$. $\widetilde{G_4^{16}}$ est alors ainsi :



Question 3 *Calculer*

a) *pour* G_4^{16}, $C([deg(s_0), deg(s_1), ..., deg(s_3)])$,

b) *pour* G_{1000}^{16}, $C([deg(s_0), deg(s_1), ..., deg(s_{999})])$,

c) *pour* $G_{10\,000}^{16}$, $C([deg(s_0), deg(s_1), ..., deg(s_{9\,999})])$

Question à développer pendant l'oral 2 *Discutez le choix de structure de données pour représenter ces graphes en mémoire.*

Note : pour le u_0 et le $\widetilde{u_0}$ fournis, tous les graphes G_n^a étudiés sont connexes, on pourra donc utiliser cette propriété pour simplifier les algorithmes.

2 Partitionnement : Cas général

Dans cette partie, on suppose que le nombre de sommets du graphe manipulé est pair. On souhaite alors partitionner ce graphe en deux parties A et B ayant exactement le même nombre de sommets. On appelle <u>coupe</u> l'ensemble des arêtes dont les extrêmités sont l'une dans A, l'autre dans B. On cherche à minimiser le nombre d'arêtes de la coupe, appelé <u>poids</u> de la coupe.

2.1 Heuristique itérative

On utilise d'abord une approche heuristique qui raffine progressivement la coupe obtenue.

- On part du partitionnement $A = \{s_0, s_1, ..., s_{\frac{n}{2}-1}\}$ $B = \{s_{\frac{n}{2}}, s_{\frac{n}{2}+1}, ..., s_{n-1}\}$
- À chaque étape, on cherche une paire de sommets $(s_i, s_j) \in A \times B$, telle qu'échanger les deux sommets s_i et s_j entre A et B diminue le plus possible le poids de la coupe (la différence étant appelée <u>gain</u>). En cas d'égalité, on minimise i, puis j.
- On répète la deuxième étape tant que l'on parvient à diminuer le poids de la coupe.

Question 4 *Calculer le poids de la coupe obtenu par cette heuristique pour les graphes*
a) G_8^{16} ***b)*** G_{100}^{16} ***c)*** G_{300}^{16}

Question à développer pendant l'oral 3 *Expliquez l'algorithme utilisé pour implémenter cette heuristique, les structures de données qu'il utilise, et donnez la complexité de chaque étape.*

L'heuristique étant coûteuse, on essaie d'en réduire le coût. On choisit d'abord i tel que passer s_i de A à B produit une coupe de poids minimum. On choisit ensuite j tel que échanger s_i et s_j entre A et B produit une coupe de poids minimum.

Question 5 *Calculer le poids de la coupe obtenu par cette heuristique pour les graphes*
a) G_8^{16} ***b)*** G_{100}^{16} ***c)*** G_{300}^{16} ***d)*** G_{1000}^{16}

Question à développer pendant l'oral 4 *Donnez la nouvelle complexité de chaque étape.*

Question à développer pendant l'oral 5 *Il serait naturel de s'attendre à ce que le poids obtenu par cette seconde heuristique soit en général plus grand que celui obtenu par la première : pouvez-vous justifier cette intuition ? De fait, on constate que le poids n'est pas plus grand dans le cas des graphes générés par* $\widetilde{u_0}$ *notamment : comment peut-on expliquer ce phénomène ? (il est même souvent moindre : on ne cherchera pas à expliquer pourquoi)*

On revient sur la première approche (qui en général obtient effectivement un poids moindre), et l'on essaie de réduire son coût. Avant de choisir la paire (i, j), on élimine les candidats inintéressants de A et de B. On calcule d'abord séparément pour les sommets de A le meilleur gain que l'on peut obtenir en passant un sommet dans B, et pour les sommets de B le meilleur gain que l'on peut obtenir en passant un sommet dans A. On ne conserve alors comme candidats parmi A que les sommets dont le gain est au moins aussi grand que le meilleur gain que l'on peut obtenir en passant un sommet de A dans B, moins 2. De même, on ne conserve comme candidats parmi B que les sommets dont le gain est au moins aussi grand que le meilleur gain que l'on peut obtenir en passant un sommet de B dans A, moins 2. On peut alors chercher une paire (i, j) parmi ces candidats seulement. En cas d'égalité, on minimise i, puis j.

Question à développer pendant l'oral 6 *Pourquoi peut-on se contenter de ces candidats, et pourquoi « moins 2 » ?*

Question 6 *Vérifiez que cette heuristique obtient les mêmes poids de coupe que l'heuristique de la question 4.*
Calculer le poids de la coupe obtenu par cette heuristique pour les graphes
a) G^{16}_{500} ***b)*** G^{16}_{1000} ***c)*** G^{16}_{2000} *(ce dernier peut prendre plusieurs secondes à calculer)*

2.2 Algorithme exact

On cherche désormais à obtenir une coupe de poids minimal. On note $min(n, a)$ le poids minimal de la coupe du graphe G^a_n en deux parties.

Question 7 *Calculer*

a) $min(8, 16)$,

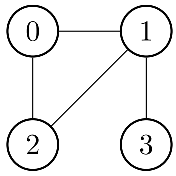
b) $C([min(16, 0), min(16, 1), ..., min(16, 9)])$,

c) $min(24, 16)$

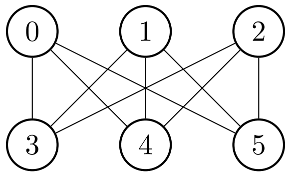
Question à développer pendant l'oral 7 *Expliquez l'algorithme utilisé et les structures de données qu'il utilise, estimez sa complexité.*

3 Cas des graphes planaires

Un graphe <u>planaire</u> est un graphe tel que l'on peut le dessiner dans le plan sans que des arêtes se croisent. Par exemple, $\widetilde{G^{16}_4}$ vu précédemment est planaire :



Alors que ce graphe-ci :



est bien connu [1] pour n'être pas planaire : on ne peut le dessiner dans le plan sans qu'au moins deux arêtes se croisent.

3.1 Génération de graphes planaires

On souhaite générer des graphes qui sont planaires par construction. Pour cela on garde note, au cours de la génération, d'un ensemble de sommets appelé contour, autour duquel on ajoute les nouveaux sommets.

On construit le graphe P_n^a à n sommets $\{s_0, s_1, ..., s_{n-1}\}$ progressivement ainsi :

- Initialement, le graphe ne contient que trois sommets s_0, s_1, s_2, les arêtes (s_0, s_1), (s_0, s_2), (s_1, s_2), et le contour est composé des trois sommets $C_0 = \{s_0, s_1, s_2\}$
- À l'étape i ($i \geq 0$), on considère le sommet du contour C_i d'indice minimal, qui est s_i. On considère également

$$d_i = \begin{cases} 5 \text{ quand } u(a+i) \bmod 100 \leq 30 \\ 7 \text{ quand } u(a+i) \bmod 100 = 31 \\ 6 \text{ quand } 31 < u(a+i) \bmod 100 \end{cases}$$

  On pose alors

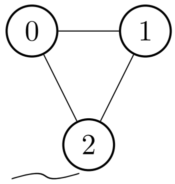
$$d'_i = min\,(max(d_i - deg(s_i), 1), n - n_i)$$

  qui est le nombre de sommets que l'on va ajouter au graphe, où n_i est le nombre de sommets actuels (qui est égal à $i + |C_i|$). On s'assure ainsi d'ajouter au moins un sommet, tout en s'assurant qu'on ne dépasse jamais le nombre n de sommets ciblé.
- On ajoute donc d'_i sommets $\{s_{n_i}, s_{n_i+1}, ..., s_{n_i+d'_i-1}\}$.
- On ajoute également les arêtes allant du sommet s_i à ces sommets : (s_i, s_{n_i}), (s_i, s_{n_i+1}), ... $(s_i, s_{n_i+d'_i-1})$
- On considère également les deux sommets s_x et s_y ($x < y$) qui sont voisins de s_i dans le graphe et appartiennent au contour C_i (par construction, s_i a toujours exactement deux voisins dans le contour). On ajoute les arêtes pour former un chemin de s_x à s_y passant par les nouveaux sommets : (s_x, s_{n_i}), (s_{n_i}, s_{n_i+1}), ..., $(s_{n_i+d'_i-1}, s_y)$.
- Enfin, on retire le sommet s_i du contour, et l'on y ajoute les nouveaux sommets à la place : $C_{i+1} = (C_i \setminus \{s_i\}) \cup \{s_{n_i}, s_{n_i+1}, ..., s_{n_i+d'_i-1}\}$
- Si le nombre de sommets a atteint n, l'algorithme est terminé, sinon on passe à l'étape $i+1$.

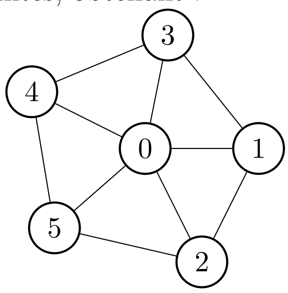
Par exemple, pour pour $u_0 = \widetilde{u_0}$, la construction de $\widetilde{P_8^{32}}$ s'effectue ainsi :

- Initialement on a

1. Il s'agit de l'énigme des trois maisons.

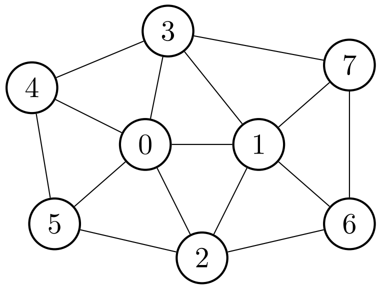


— On effectue l'étape $i = 0$. Puisque $u(32+0) \bmod 100 = 14$, $d_0 = 5$. s_0 étant de degré 2 pour l'instant, on obtient $d'_0 = 3$. Les voisins de s_0 (dans le graphe) qui sont dans le contour $C_0 = \{s_0, s_1, s_2\}$ sont (dans l'ordre) s_1 et s_2. On ajoute ainsi trois sommets s_3, s_4 et s_5 et les arêtes correspondantes, obtenant :



Et le nouveau contour est $C_1 = \{s_1, s_2, s_3, s_4, s_5\}$.

— On effectue l'étape $i = 1$. Puisque $u(32+1) \bmod 100 = 44$, $d_1 = 6$. s_1 est de degré 3 pour l'instant, mais l'on a déjà $n_1 = 6$ sommets or on cible n sommets, on obtient donc $d'_1 = 2$. Les voisins de s_1 (dans le graphe) qui sont dans le contour C_1 sont (dans l'ordre) s_2 et s_3. On ajoute ainsi les deux derniers sommets s_6 et s_7 et les arêtes correspondantes, obtenant :



Et le dernier contour est $C_2 = \{s_2, s_3, s_4, s_5, s_6, s_7\}$.

Question à développer pendant l'oral 8 *Quel est à peu près le degré moyen des sommets des graphes obtenus ? Quel type de structure de données vaut-il mieux utiliser pour représenter ces graphes en mémoire ?*

Question 8 *Calculer*

a) *pour* P^{32}_{16}, $C([deg(s_0), deg(s_1), ..., deg(s_{15})])$,

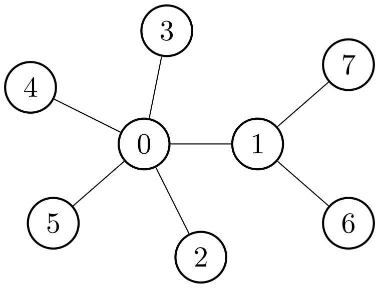
b) *pour* P^{32}_{1000}, $C([deg(s_0), deg(s_1), ..., deg(s_{999})])$,

c) *pour* $P^{32}_{50\,000}$, $C([deg(s_0), deg(s_1), ..., deg(s_{49\,999})])$

3.2 Arbre couvrant

Pour les graphes planaires générés ci-dessus, on souhaite extraire un arbre couvrant, c'est-à-dire un graphe qui comprend les mêmes sommets, mais un sous-ensemble des arêtes de façon à produire un arbre. Pour cela, on effectue un parcours en largeur du graphe, et l'on ne conserve dans l'arbre couvrant que les arêtes utilisées lors du parcours en largeur. On parcourt à chaque fois les sommets et les voisins par ordre croissant d'indice dans le graphe.

On note $T_n^a(i)$ l'arbre couvrant ainsi obtenu pour le graphe P_n^a en partant du sommet s_i.
Par exemple, pour $u_0 = \widetilde{u_0}$, pour le graphe planaire $\widetilde{P_8^{32}}$, en partant du sommet s_0, on parcourt ses voisins et ajoute ainsi à l'arbre couvrant dans l'ordre les arêtes (s_0, s_1), (s_0, s_2), (s_0, s_3), (s_0, s_4), et (s_0, s_5). On repart alors successivement de ces différents sommets par ordre croissant d'indice, d'abord avec s_1, pour lequel on ajoute les arêtes (s_1, s_6) et (s_1, s_7). Puis pour s_2, s_3, s_4, s_5 il n'y a pas d'arête à ajouter. On repart alors successivement des nouveaux sommets s_6 et s_7, pour lesquels il n'y a pas d'arête à ajouter. On obtient ainsi l'arbre couvrant $\widetilde{T_8^{32}(0)}$:



Question 9 *Calculer*

a) *dans* $T_{16}^{32}(0)$, $C([deg(s_0), deg(s_1), ..., deg(s_{15})])$,

b) *dans* $T_{1000}^{32}(0)$, $C([deg(s_0), deg(s_1), ..., deg(s_{999})])$,

c) *dans* $T_{50\,000}^{32}(0)$, $C([deg(s_0), deg(s_1), ..., deg(s_{49\,999})])$

3.3 Partitionnement de graphes planaires

Dans le cas des graphes planaires, on change légèrement le problème de partitionnement : pour un graphe G ayant n sommets on cherche un sous-ensemble S des sommets (appelé séparateur) tel que si l'on retire ces sommets du graphe G et les arêtes correspondantes, on obtient deux ensembles de sommets A et B pour lesquels il n'y a pas dans G d'arête dont une extrémité est dans A et l'autre dans B. S « sépare » donc effectivement le graphe en deux parties. On cherche à réduire le nombre de sommets du séparateur, $|S|$, tout en gardant A et B de tailles similaires.

Question à développer pendant l'oral 9 *En pratique, on constate que l'on parvient en général à obtenir* $|S| = O(\sqrt{n})$*, pourquoi ?*

Nos graphes planaires étant très réguliers, nous allons pouvoir utiliser une heuristique un peu simpliste pour déterminer S, mais qui se révèlera obtenir de bons résultats.

— On commence par calculer un arbre couvrant tel que réalisé à la section 3.2, en partant du sommet 0.
— On calcule également les profondeurs des différents sommets dans cet arbre, on note $maxp$ la profondeur maximum obtenue.
— On note H l'ensemble des sommets de profondeur au moins $maxp - 1$.
— On pioche s_i le sommet de H avec i maximal.
— On choisit $s_j \in H$ tel que le chemin de s_i à s_j dans l'arbre couvrant forme un séparateur S pour lequel $||A| - |B||$ est minimal, en choisissant j minimal en cas d'égalité.

Note : même si ce n'est pas vrai dans le cas général, dans le cas des graphes planaires générés mentionnés dans cet exercice, il se trouve que cette heuristique les partitionne toujours en au plus 2 composantes connexes.
On note S_n^a le séparateur obtenu ainsi pour P_n^a.

Par exemple, pour $u_0 = \widetilde{u_0}$, dans le cas du graphe $\widetilde{P_8^{32}}$, H comprend tous les sommets sauf s_0, on part du sommet s_7, et l'on obtient $\widetilde{S_8^{32}} = \{s_0, s_1, s_5, s_7\}$ pour lequel $||A| - |B|| = |2 - 2| = 0$

Question 10 *On ordonne les sommets de S_n^a par indice croissant. Calculez*
a) $C(S_{16}^{32})$ ***b)*** $C(S_{1000}^{32})$ ***c)*** $C(S_{10\,000}^{32})$

Question à développer pendant l'oral 10 *Expliquez votre implémentation et estimez sa complexité.*

◇
◇ ◇
◇

Fiche réponse <u>type</u> : Comment couper la poire en deux ?

$\widetilde{u_0}$: 42

Question 1

a) 781

b) 374

c) 805

Question 2

a) 373

b) 167

c) 943

Question 3

a) 33

b) 233

c) 506

Question 4

a) 2

b) 341

c) 1915

Question 5

a) 2

b) 327

c) 1900

d) 12511

Question 6

a) 4314

b) 12572

c) 36952

Question 7

a) 2

b) 386

c) 20

Question 8

a) 870

b) 348

c) 582

Question 9

a) 687

b) 235

c) 377

Question 10

a) 111

b) 375

c) 923

Fiche réponse : Comment couper la poire en deux ?

Nom, prénom, u_0 : ....

Question 1

a)

b)

c)

Question 2

a)

b)

c)

Question 3

a)

b)

c)

Question 4

a)

b)

c)

Question 5

a)

b)

c)

d)

Question 6

a)

b)

c)

Question 7

a)

b)

c)

Question 8

a)

b)

c)

Question 9

a)

b)

c)

Question 10

a)

b)

c)